Movie Recommendations System Using Ml

[1]Shivam Sharma,
Department of electronics & communication Engineering,
KIET Group of Institution's Ghaziabad, India

[2]Shubham Shukla,
Department of electronics & communication Engineering,
KIET Group of Institutions Ghaziabad, India

[3]Shikhar Gaur,
Department of electronics & communication Engineering,
KIET Group of Institutions Ghaziabad, India

[4]Vibhav kumar Sachan,
Department of electronics & communication Engineering,
KIET Group of Institutions Ghaziabad, India

Abstract:- A recommendation engine is a type of data filtering technology that uses machine learning techniques to provide the most relevant recommendations to a particular user or client. It operates by searching for patterns in client behavior data, which may be collected explicitly or implicitly. It first records a client's area of interest and then uses that information to enable product recommendations for customers who appear tobe shopping. For example: An e-commerce website won't know anything about a visitor if they are completely new to it. So how would the positioning strategy advocate the product to the consumer in this situation? One practicalsolution may be to suggest the item that is in great demand, or the one that is popular. Another practical option is to recommend the product that will likely bring more profit to the company. Recommendation engines can be implemented by using 3 strategies: - Collaborative filtering (focuses on collecting and analyzing data about user behavior, preferences, and activities to predict what a person would like based on how they are similar to other users.), content-based filtering (which works on the principle that if you like one thing, you'll like this other thing, too) and hybrid models.

Keywords: Recommendation System,Collaborative Filtering Approach, And Content- Based Filtering Method.

I. INTRODUCTION

Recommendation algorithms are present everywhere these days. Several of the biggest IT companies including Netflix, Amazon, Flipkart, Google engage with everyday use recommendation systems.

Recommender systems are commonly used in various fields including movies, music, news, books, articles and various other products.

The main objective of this system is to filter the data and provide the result to user in terms of a particular item with specified domain. The domain- specific item for this particular system is movies, so our recommendation system mainly focuses on filtering and predicting the movies based on the interests and data provided by the user. There are several methods to develop a movie recommendation engine, but we built ours using a content-based filtering strategy. With the help of content-based filtering our system providesthe result and various other options like the user's interest.

For ex:- if a user wishes to watch a superhero movie, then the recommendation system will recommend all top movies related to superheroes.

II. RELATED WORK

The research of recommender systems has expanded greatly during the past few years. Several methodologies are used by recommendation systems to provide useful recommendations. Both collaborative filtering and content-based filtering are typically utilized. A user is given a recommendation for an item via a content- based filtering system based on their own preferences.

There are several movie recommendation algorithms available nowadays. To provide the most accurate suggestions, the developers of these systems collect data from social media to determine consumerpreferences.

In this work, we provide a recommendation system that evaluates the user's preferences to suggest the best option. The quality and content are considered via collaborative filtering and content-based filtering. Our goal is to deliver suggestions to users by taking the search experience into careful, organized consideration.

- *Technique Used*

Our movie recommendation engine employs content-based filtering as its method. The ratings and reviews that users submit are used in a content-based movie recommendation system. A user profile is constructed using this data, and it is then used to suggest movies to both the existing user and new users.

This filtering technique provides more accurate and better results if the user gives more inputs on the recommendation system.

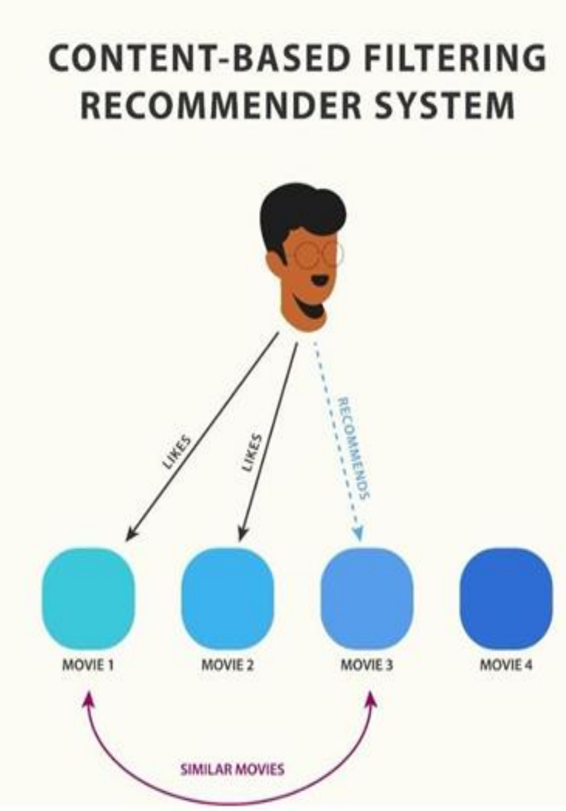


Fig 1 Content Based Filtering

Content-based method filtering utilizes Cosine angle similarity and Term frequency to get the data as shown in figure 1.

- *Term Frequency (Tf)*

Term Frequency refers to the frequency ofa particular word in the data. TF is a statistical measure which determines howrelevant a word is to a document in thedatafile.

The weight or occurrence of a word in a document can be counted by using the equation below:

$$w_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tf}_{t,d}, & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

Fig 2 Formula for Filtering the Data

- *Cosine Angle Similarity*

Cosine angle similarity is a numeric value which ranges between 0 to 1. It is used to determine the similarity of two different items from 0 to 1 in regard for one another.

This score is calculated by comparing the texts of the two documents in the data. Cosine Similarity can be measured betweentwo texts irrespective of their sizes.

The cosine angle between two vectors projected in multi-dimensional space is determined by cosine similarity.

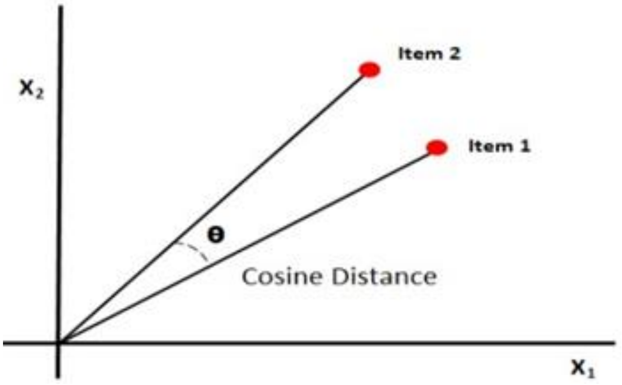


Fig 3 Cosine Distance

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Fig 4 Formula for Finding Similarity among the Data

This formula aids in determining whether two vectors projected on a multi-dimensional planeare identical.

- *Proposed System*

Basic architecture of our workflow in making this system is:

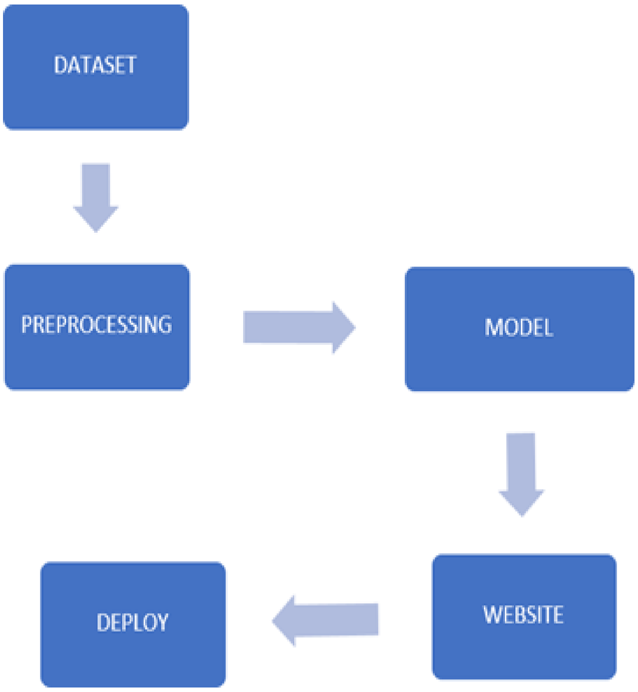


Fig 5 Methodology Used

The brief description on above blocks are as follows:

- *Dataset*

The dataset is the most crucial component of a machine learning process. So, in our movie recommender system we have used “TMDB 5000 Movie Dataset”. This dataset contains Hollywood movies released on or before January 2018.

This data consists of cast, crew, genre, plot keywords, budget, revenue, posters, releasedates, popularity.

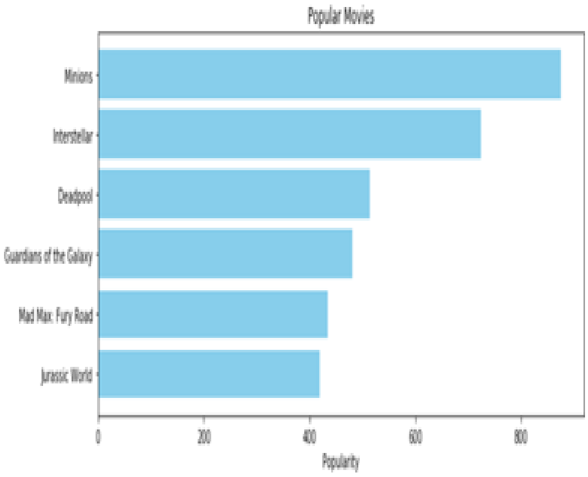


Fig 6 Graph of Movies Dataset Based on Popularity

We may also use ratings to represent the dataset; to create the graphic, we utilize IMDB's weighted rating system.

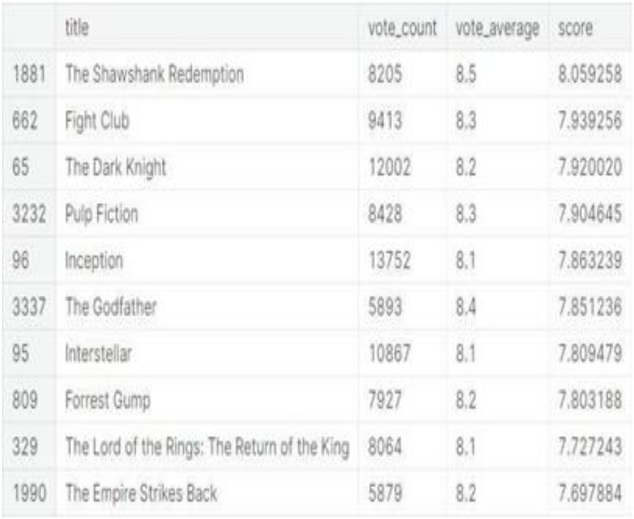
The equation is represented as:

$$Weighted\ Rating(WR)=(\frac{v}{v+m}.R)+(\frac{m}{v+m}.C)$$

Fig 7 Formula for Calculating WR

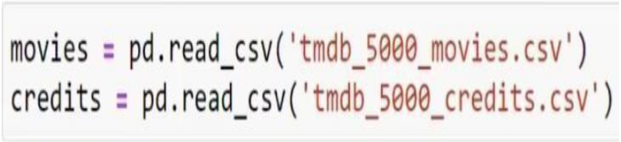
✓ *Above Variable Represents:*

- v represents vote totals for the film.
- m represents minimal number of votes needed to get seen in the chart.
- R represents mediocre rating of film C represents mean vote.



	title	vote_count	vote_average	score
1881	The Shawshank Redemption	8205	8.5	8.059258
662	Fight Club	9413	8.3	7.939256
65	The Dark Knight	12002	8.2	7.920020
3232	Pulp Fiction	8428	8.3	7.904645
96	Inception	13752	8.1	7.863239
3337	The Godfather	5893	8.4	7.851236
95	Interstellar	10867	8.1	7.809479
809	Forrest Gump	7927	8.2	7.803188
329	The Lord of the Rings: The Return of the King	8064	8.1	7.727243
1990	The Empire Strikes Back	5879	8.2	7.697884

Fig 8 Calculated Weighted Rating for the Movies in the Dataset



```
movies = pd.read_csv('tmdb_5000_movies.csv')
credits = pd.read_csv('tmdb_5000_credits.csv')
```

Fig 9 Syntax of Linking the Dataset with the System Code

- *Preprocessing*

In machine learning, we use pandas and NumPy libraries for preprocessing.

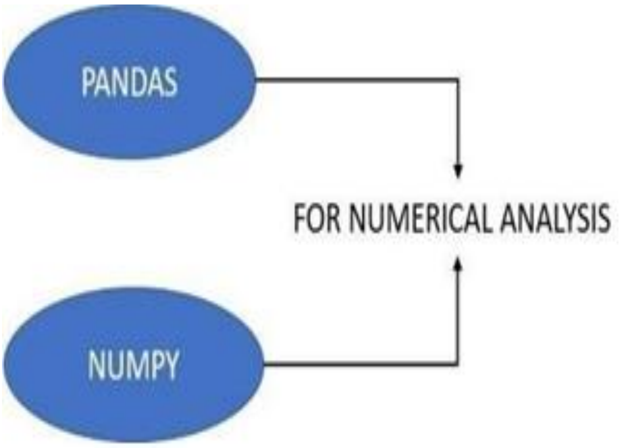
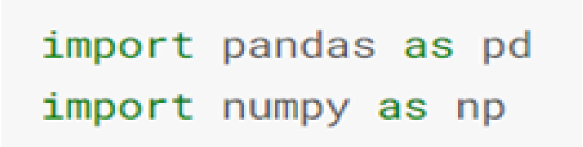


Fig 10 Numerical Analysis

Numeric Python is referred to as NumPy. It is a library for working with arrays in Python.

Moreover, it may be used with algebra, matrices, and the Fourier transform.

Python's Pandas package is used to manipulate datasets. Data analysis, cleansing, exploration, and manipulation are performed by this library function.



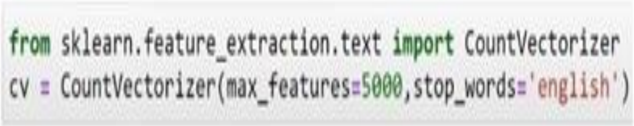
```
import pandas as pd
import numpy as np
```

Fig 11 Libraries Used

The most practical and reliable machine learning library is Scikit-learn (Sklearn). This Python library was created using Numpy, Scipy, and Matplotlib.

Sklearn assists with graph representation of the datasets..

A collection of text documents is transformed into a matrix of token counts using the CountVectorizer function from the Sklearn toolkit.



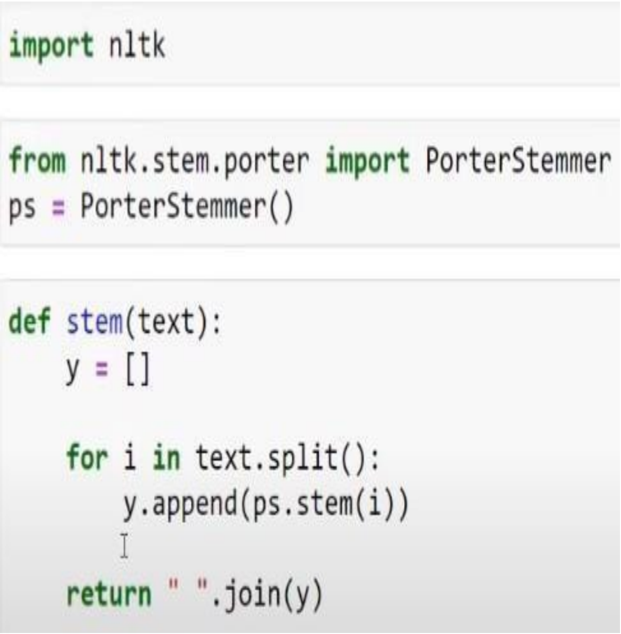
```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=5000,stop_words='english')
```

Fig 12 CountVectorizer Syntax

CountVectorizer returns Scipy sparse matrix and this matrix is then converted into a Numpy array.

Stemming or NLTK it is the process of reducing a word into its base form by removing affixes from them. For example data [dance, dancing] will be reduced to [dance, dance] with help of stemming.

For stemming process **NLTK** library is used which stands for National language toolkit.

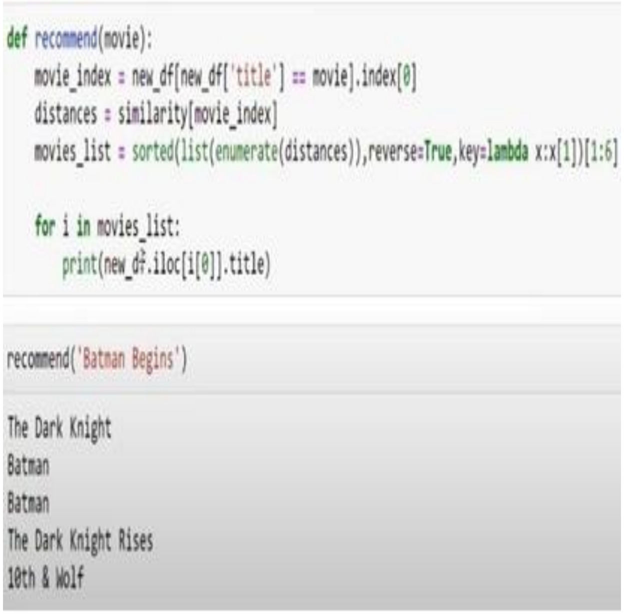


```
import nltk

from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()

def stem(text):
    y = []

    for i in text.split():
        y.append(ps.stem(i))

    return " ".join(y)
```

Fig 13 NLTK Library Syntax

- *Model*

After performing all the preprocessing steps the recommendation model is completed.



```
def recommend(movie):
    movie_index = new_df[new_df['title'] == movie].index[0]
    distances = similarity[movie_index]
    movies_list = sorted(list(enumerate(distances)),reverse=True,key=lambda x:x[1])[1:6]

    for i in movies_list:
        print(new_df.iloc[i[0]].title)

recommend('Batman Begins')

The Dark Knight
Batman
Batman
The Dark Knight Rises
10th & Wolf
```

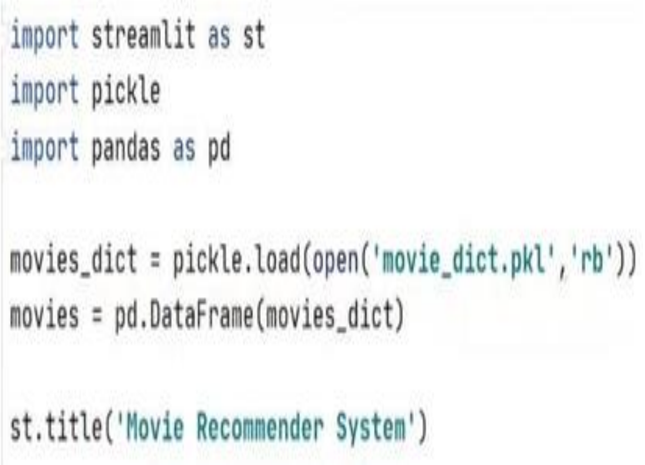
Fig 14 Recommendation System Code

Hence, this recommend function returns the top recommended movies after passing the title of the movie as an input.

As we have passed 'Batman Begins' as the input the recommendation model has given us recommended movies related to Batman in the form of output.

- *Website*

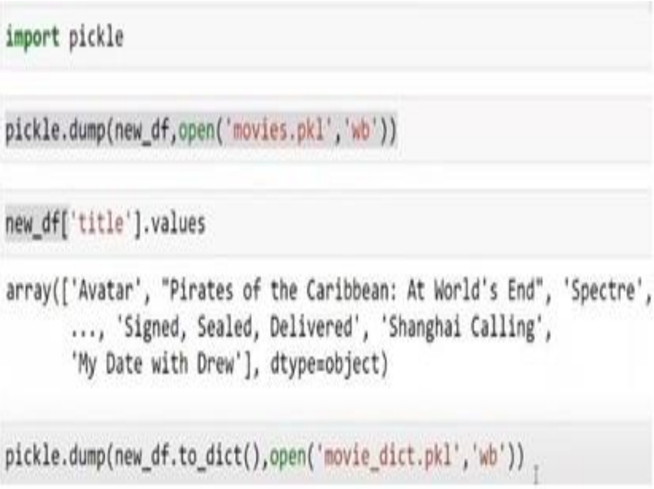
In this step the recommendation model is converted into a website with the help of python IDE pycharm. Streamlit library of python is used for the creation of this website.



```
import streamlit as st
import pickle
import pandas as pd

movies_dict = pickle.load(open('movie_dict.pkl','rb'))
movies = pd.DataFrame(movies_dict)

st.title('Movie Recommender System')
```

Fig 15 Pickle Library Syntax

Pickle library is used to connect recommendation system modelwith website.



```
import pickle

pickle.dump(new_df,open('movies.pkl','wb'))

new_df['title'].values

array(['Avatar', "Pirates of the Caribbean: At World's End", 'Spectre',
       ..., 'Signed, Sealed, Delivered', 'Shanghai Calling',
       'My Date with Drew'], dtype=object)

pickle.dump(new_df.to_dict(),open('movie_dict.pkl','wb'))
```

- *Deploy*

This website is deployed on the Heroku server. Three files are created in order to run the Streamlit app on the Heroku server those three files are 'procfile', 'steup.sh' and gitignore.

"pip freeze > requirements.txt" command is used to generate all the required libraries to run the website on Heroku server.

III. RESULTS

After the implementation and completion of all the steps the model is tested to check if its working properly or not.

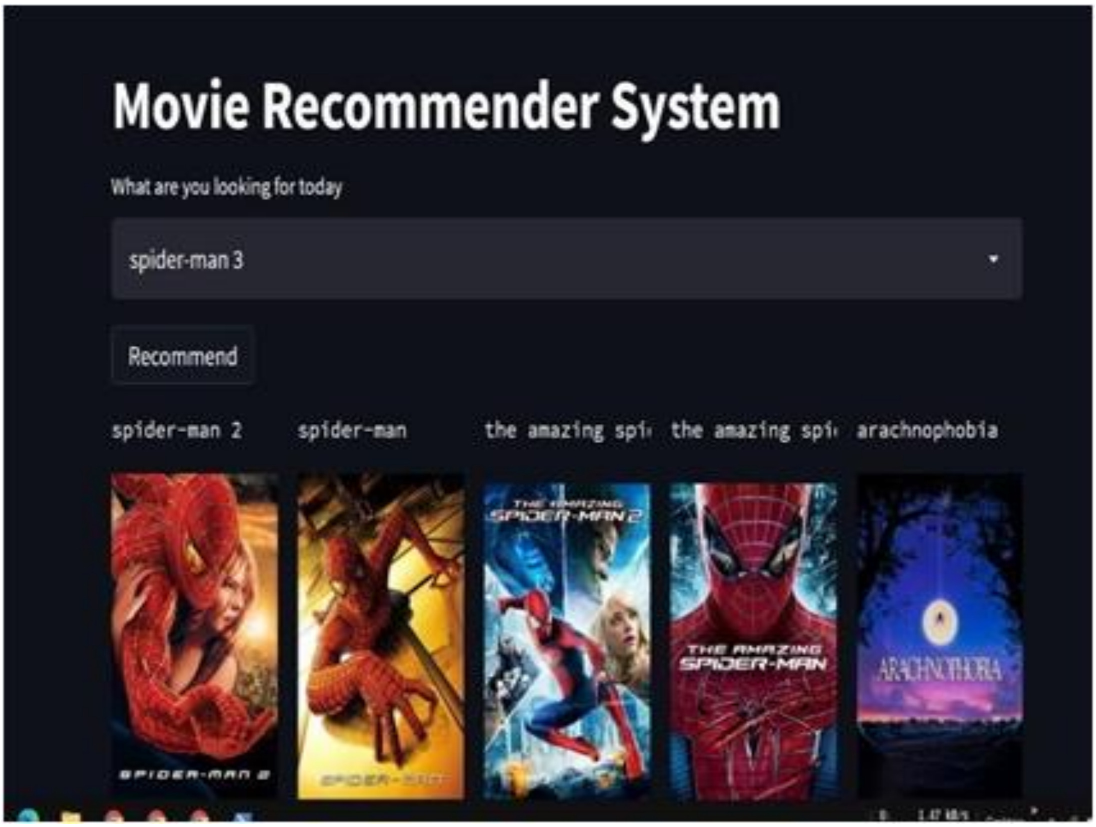


Fig 16 Homepage

This is the homepage (landing page) of our website user can type the name of the movie in search box for getting recommendations.

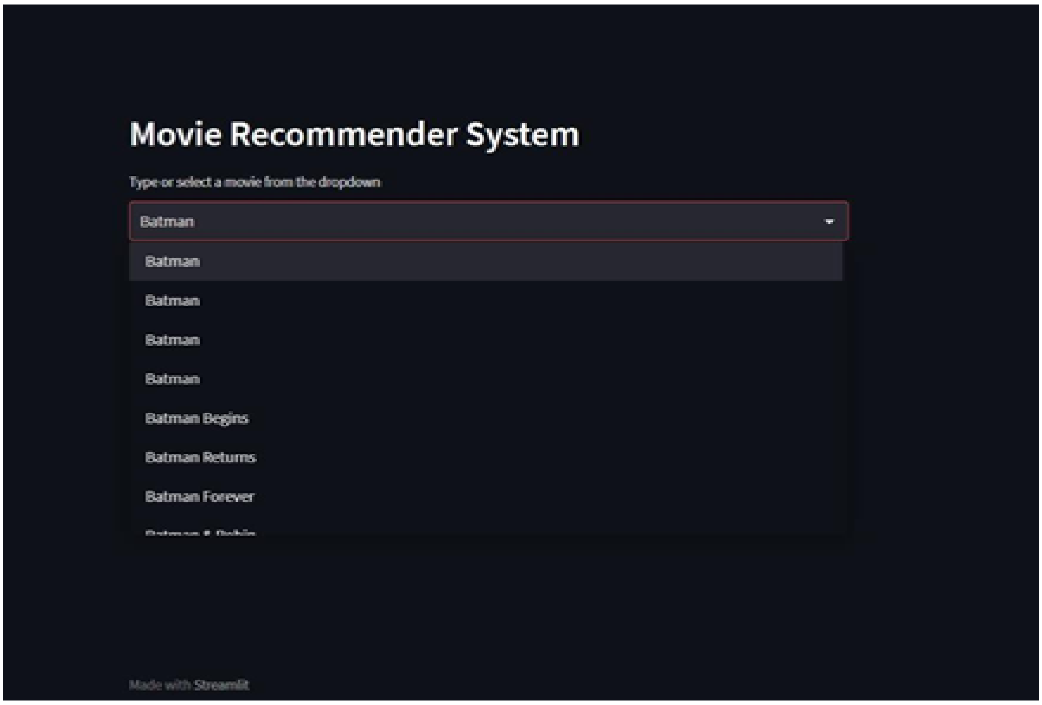
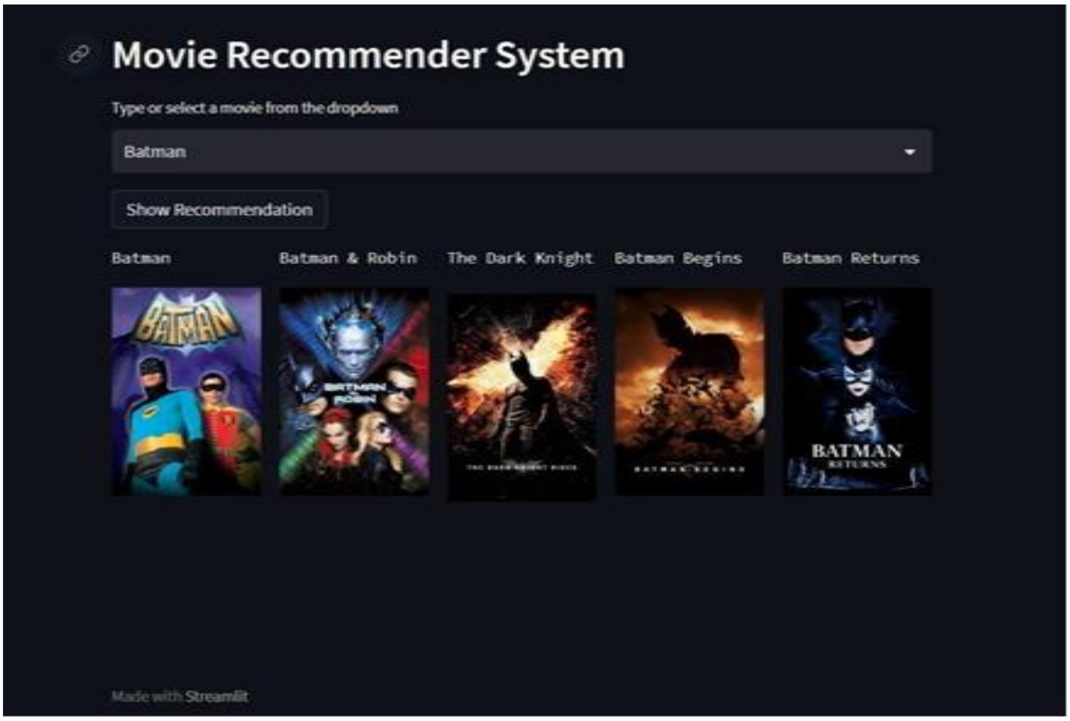


Fig 17 Search

Here, user is searching for a movie named “Batman”.



Movie Recommender System

Type or select a movie from the dropdown

Batman

Fig 18 Result (Recommended Movies)

As we can see, user got the recommendations related to movies named “Batman”.

IV. CONCLUSION

This study describes a content-based filtering- based approach for making movie recommendation system. Making the recommendation system effective was the main goal. The main objective is to create a system that can offer users high-quality suggestions without needing a lot of personal information, search history, etc. The results of the trials show that the recommended technique provides consumers with relevant suggestions.

- *Future Enhancements*

Future work will involve using user ratings of movies on websites like Rotten tomatoes, IMDb, etc. We will try to use Hybrid filtering technique in our model to get better predictions and recommendations.

Furthermore, we will try to add the data of the movies from various film industries.

We will also work on improving and updating theuser interface of our website in order to make our website user friendly.

ACKNOWLEDGEMENT

The authors would like to express gratitude towards KIET Group of Institutions and Dr. Shubham Shukla sir (ECE Dept.) fortheirfinancial and technical guidance.

REFERENCES

[1]. Bilge, A., Kaleli, C., Yakut, I., Gunes, and Polat: A study on collaborative filtering techniques. International Journal of Software Eng. Knowl. Eng. 23(08), 1085–1086 (2013) CrossRef with Google Scholar.

[2]. Regarding the identification of fake binary ratings: Okkalioglu, M., Koc, M., Polat, H. SAC 2015, pp. 901–907 in Proceedings of the 30th Annual ACM Symposium on Applied Computing. America, ACM (2015).

[3]. Kaleli, C., and Polat, H.: Privacy-preserving naive bayesian classifier based upon on distributed data. Computer Intelligence 31(1):47-68 (2015)

[4]. Hongli Lin, Xuedong Yang, and Weisheng Wang. A content-boosted collaborative filtering algorithm for personalized training in interpretation of radiological imaging. Journal of digital imaging. 107-13 (2014)

[5]. Harpreet Kaur Virk, Er Maninder Singh, and A Singh. Analysis and design of hybrid online movie recommender system. International Journal of Innovations in Engineering and Technology Volume, 5,2015.

[6]. Urszula Ku zelewska. Recommendation system engines. Iranian Journal of Energy and Environment, 2019.